



**COMPUTER
CENTRE**

BULLETIN

*Volume 2, Number 5.
6th May, 1969.*

*Editor:
H. L. Smythe.*

PDP 10 ON THE AIR

The Computer Centre is pleased to be able to announce to clients that the PDP 10 Operating System is now available for general use. The PDP 10, one of the fastest computers in Australia, is a dual processor, general purpose, digital computer which has a magnetic core store of 48K words.

In this early stage, the service offered on the PDP 10 has several limitations. However, these restrictions will be gradually relaxed as soon as the necessary systems programming and proving tests are completed. Initially, only a *batch processing* service (similar to that on the GE 225) is available. Remote terminals will not be introduced at this time. While other programming languages will be available later, *FORTRAN IV*, only, will be used initially. Input/output are restricted to the *card reader* and the *line printer*. Pending the availability of a complete file storage system, the use of the disk as a scratch device will be allowed as soon as possible. *Binary object programs* will not be produced i.e. a program will have to be compiled each time it is run. As compilation on the PDP 10 is very fast, this should not be a serious restriction. Object programs will be limited in size to a *maximum of 24K words*. Only this restriction will be permanent. As a primary objective in the development of the PDP 10 is that all programs written during the initial period must be compatible with the system at a later date, these limitations are both worthwhile and necessary.

THIS EDITION

This issue of the Bulletin is devoted to assisting clients of the Computer Centre to use the PDP 10 with a minimum of difficulty. Two articles are included, one detailing the control cards used in batch processing on the PDP 10, and the other giving helpful advice on the conversion from GE 225 to PDP 10 FORTRAN IV. As well, several errors in the PDP 10 FORTRAN IV compiler are described for clients' reference. This edition also introduces some new members of staff, and lists the Library Accessions for January. Readers

interested in the PDP 10 computer system should see the April Bulletin for information on the charges for the PDP 10 services, and for a description of FORTRAN memory allocation on the PDP 10.

STAFF OF THE COMPUTER CENTRE

The Computer Centre welcomes with pleasure several new members of staff.

A SENIOR DEMONSTRATOR

Leonard (Len) Mor has been appointed temporary Senior Demonstrator, and is currently working on accounting programs for the PDP 10 Batch Operating System. Len graduated with Honours in Electrical Engineering in 1964, becoming a Communications Engineer with the P.M.G. In 1967, he completed the programming section of the Diploma in Automatic Computing, and, in the following year, submitted his thesis for a Master of Engineering Science degree. The thesis was concerned with the development and implementation of a language for using analogue computer techniques on a digital computer.

.... AND A PURCHASING OFFICER

The Computer Centre has also acquired a Purchasing Officer, a position created to reduce the work load on the Maintenance Technicians, Graham Jerrard and Doug Brunner. This has largely been caused by the addition of the PDP 10 computer system. The Purchasing Officer, *Dal Anderson*, is responsible for maintaining adequate supplies of all consumable stores such as punch cards and data pads, and supervises all aspects of purchasing, issuing orders, checking deliveries, and passing related accounts for payment. Dal will also maintain the Equipment and Furniture Register for the Department of Computer Science.

.... AND THE SECRETARY

The former Secretary, Delphine Dare, who has departed for the colder shores of England, has been replaced by *Mrs. Judith Greenhill*. Judith has been working at the University since November 1962, and was appointed Head Typist of Staff Section, in charge of six typists, in January 1965. In March this year, she became Secretary of the Department of Computer Science.

LIFE ON THE OCEAN WAVE

John Row, one of our Systems Programmers, has two months' leave from the Computer Centre. As assistant navigator on the yacht "Kaleena", he will take part in the Auckland to Suva race. Providing there are no errors in his program, and the sharks are friendly, John should be back at his desk on 2nd June.

PROGRAMMING ADVICE

CONVERTING FROM GE 225 TO PDP 10 FORTRAN IV - J.S. Williams

Now that the PDP 10 FORTRAN IV Batch system is operational, many clients will wish to compile and run programs originally written for the GE 225. Problems in converting programs according to the joint FORTRAN IV manual (Reference 1) should be minimal. However, programs written prior to the publication of that manual will have to be converted. This article* describes only the differences which may prevent the compilation and/or execution of a GE 225 FORTRAN IV program on the PDP 10. It does not treat the advanced features (Reference 2) or operating procedures (Reference 3).

* The article is based on reference 4, copies of which can be obtained from the Computer Centre.

Each time a program is to be converted, carefully check the following questions. If the answer to a particular question is "No", that comment can be ignored.

1. Do any real variables or constants have values, in magnitude, outside the range .14E-38 to 1.7E+38?

Real variables and constants for the PDP 10 must lie in this range. If they do not, the program must be modified.

2. Are there any arithmetic IF statements with only 2 statement numbers?

Arithmetic IF statements in PDP 10 FORTRAN IV must have all 3 statement numbers, e.g.

IF(N-3)10,20,30

3. Does the program manipulate *characters*?

Care must be taken in converting programs that manipulate alphanumeric fields. On the PDP 10, integer and real variables can both have a maximum of 5 characters, since one location is used in both cases. Double precision and complex require 2 locations each, and hence can store a maximum of 10 characters.

The GE 225 can have a maximum of 3 characters for an integer variable, 6 for a real variable and 12 for double precision and complex variables.

4. Are there any PRINT statements where the first character to be printed on a line is neither a format control character nor a blank?

FORTRAN IV on the PDP 10 *never prints the first character* in a line of output. It interprets it as a format control character. If the character is not a control character, it has the same effect as a blank, i.e. the PRINT starts on a new line.

5. Are there any FORMAT statements with fewer format specifications than variables in the input/output statement, so that a portion of the FORMAT statement has to be scanned again?

If a FORMAT statement, on the PDP 10, does not contain sufficient format specifications for the variable list, the FORMAT is repeated from the last open parenthesis of level one (or zero if there is no level one).

e.g. `FORMAT (F5.2,(2(E12.5,E10.3),3(I4,F10.6)))`

↑	↑ ↑	↑
↑	↑ ↑	↑
level 0	level 1	level 2
1	2	2

If there are more than 11 variables to be input/output, the first 11 are input/output as the first record according to the complete FORMAT statement. Second and successive records are input/output under the FORMAT:

`(2(E12.5,E10.3),3(I4,F10.6))`

6. Do any FORMAT statements use X, rather than IX, to omit one character position?

In PDP 10 FORTRAN IV, X is not interpreted as IX. It produces a compilation error message.

7. Are there any DATA statements in the program?

A DATA statement for the PDP 10 must have a comma after any intermediate slashes that follow data strings.

e.g. `DATA list 1/data-string/,list 2/data-string/`

8. Is there an EQUIVALENCE statement equivalencing (array) variables of a different mode, e.g. integer and real?

Keep in mind the different number of locations on the two computers for the different modes.

	<u>GE 225</u>	<u>PDP 10</u>
Integer	1 location	1 location
Real	2 locations	1 location
Double precision	4 locations	2 locations
Complex	4 locations	2 locations

9. Does the program use magnetic tapes?

A program using magnetic tapes can not be converted until file storage features are incorporated in the system.

The Computer Centre Staff hope that the problems of conversion for clients will be few. If there are any queries, please feel free to arrange a consultation with a programmer by contacting the Secretary, Computer Centre, extension 688.

References

1. Technical Manual No. 1
FORTRAN IV for the GE 225 and PDP 10
J.S. Williams
2. Technical Manual No. 1 - Supplement A
Additional Features of PDP 10 FORTRAN IV
R.E. Kelly
3. Technical Manual No. 1 - Supplement B
Operating Procedures for PDP 10 FORTRAN IV
R.E. Kelly
4. Technical Manual No. 1 - Supplement C
Converting From GE 225 to PDP 10 FORTRAN IV
J.S. Williams

PDP 10 FORTRAN IV ERRORS

J.D. Noad

J.S. Williams

The Bulletin is endeavouring to keep its readers informed of any errors that are discovered in the PDP 10 FORTRAN IV compiler. In this way, it is hoped to reduce the number of conversion problems for clients. Programmers are available for consultation at specified times, and the Editor would be happy to publish and to answer any queries or programming difficulties.

Multiplying elements of two arrays that both have integer functions as subscripts gives incorrect results.

The following example would not give the desired result:

```
C = A(IND(J)) * B(JEN(K))
```

where A and B are arrays and IND and JEN are functions.

An alternative method of coding must be used, e.g.

```
D = B(JEN(K))
C = A(IND(J))*D
```

When reading/printing two or more arrays with the one READ/PRINT statement, any given index can only be used once.

The following examples will give incorrect results:

- (i) PRINT 10,(A(I),I=1,N),(B(I),I=1,M)
- (ii) DO 10 J=1,L
10 PRINT 40,(A(I,J),I=1,N),(B(K,J),K=1,M)

This can be avoided by using different indices.

```
(i)      PRINT 10,(A(I),I=1,N),(B(J),J=1,M)
(ii)     DO 10 J=1,L
          JJ=J
10 PRINT 40,(A(I,J),I=1,N),(B(K,JJ),K=1,M)
```

A point of inconsistency that may be rather disturbing, arises when there is an error in a DIMENSION statement. As a result of the error, none of the arrays in the statement is defined as an array. The problem is that compilation error messages are not produced for statements that have an array on the left hand side of an equation.

e.g. A(I,J) = 1.0

Error messages, however, are produced on input/output statements that include arrays.

e.g. PRINT 90,(A(I),I=1,N)

Values can not be input/output to/from a real variable using I type format. Similarly, E or F can not be used for an integer variable.

A compilation error is not produced, and, at execution time, the value is placed in the variable in the mode specified by the FORMAT specification instead of being converted to the mode of the variable.

LIBRARY ACCESSIONS

This section details the books and periodicals concerning the computer field, that have been acquired by Libraries of the University of Queensland in January 1969.

Blatt, John Markus. *Introduction to FORTRAN IV Programming*. 1967. (512.92 BLA, Engin. Lib.) 1968. (Q651.8 BLA, Maths. Lib.)

Yu, Chia-Peir. *Digital Filter Design Technique and Realization of Transfer and Immittance Functions By Using Digital Elements.* 1967. (Q510.783 YU, Engin. Lib.)

Scherer, Herman K. *Logic, Logical Design, and Digital Circuits.* 1967. (Q621.381958 SCH, Engin. Lib.)

Computer Graphics. 1967. (Q651.8 COM, Engin. Lib.)

Conference on Industrial Measurement Techniques for On-Line Computers, London, 1968. *Industrial Measurement Techniques for On-Line Computers.* 1968. (Q651.8 CON, Elect. Engin. Lib.)

Jensen, Randall W. and Lieberman, Mark D. *I.B.M. Electronic Circuit Analysis Program; Techniques and Applications.* 1968. (621.3815018 JEN, Elect. Engin. Lib.)

McCormick, John M. and Salvadori, Mario G. *Numerical Methods in FORTRAN.* 1964. (519.22 MACC, Elect. Engin. Lib.)

Saxon, James A. and Steyer, Wesley W. *Basic Principles of Data Processing.* 1967. (651.8 SAX, Engin. Lib.)

Krishnamoorthi, B. *The Stationary Behaviour of a Time-Sharing System under Poisson Assumptions.* 1965. (Q651.84 KRI, Engin. Lib.)

Lawson, Charles M. *Computing Facility Management Survey.* 1962. (Q651.8 LAW, Engin. Lib.)

Quade, Edward S. ed. *Systems Analysis and Policy Planning.* 1968. (658.502 QUA, Engin. Lib.)

Rigney, Joseph W. *Computer Personnel Selection and Criterion Development.* 1968. (Q658.311 RIG, Engin. Lib.)

The Application of Automation in the Process Industries. Amsterdam, 1967. (Q338.4566 APP, Engin. Lib.)

California University. University at Los Angeles. *Health Sciences Computing Facility. BMD: biomedical computer programs.* 2nd ed., 1968. (Q510.78 CAL, Engin. Lib.)

Handbook for Automatic Computation. Vol. 1. (510.78 HAN, Engin. Lib.)

Dobson, R.S. *Some Applications of a Digital Computer to Hydraulic Engineering Problems.* 1967. (Q627 DOB, Engin. Lib.)

CONTROL CARDS FOR BATCH PROCESSING ON THE PDP 10

Initially, a batch processing computer service for FORTRAN IV programs only, is available on the PDP 10. Input is from the card reader, and output is on the line printer. FORTRAN IV programs are compiled and executed under the control of the PDP 10 Operating System. The following is a description of the applicable Control Cards and a sample deck setup.

1. The *Job Identification Card* is a preprinted card on which the user must write his project number, name, a date, and signature. Each of these cards will contain a unique number which will be printed on the first and last pages of output, thus serving to identify uniquely that particular run. One such card must be supplied with each run on the computer. It will be removed and retained by the Computer Centre.
2. The *Job Card* must be keypunched by the user, and contains information pertinent to that job. It contains the characters ",EQJ" in columns 1 through 4. The remaining parameters may be punched in freefield format, thus:

.JOB PROJECT=*project-number*, NAME=*surname-of-authorized-user*,
COST=\$*dollars.cents*, TYPE=*type-of-run*

where *project-number* is the project to which the cost of the run is to be charged;

surname-of-authorized-user is the surname of the user designated as being responsible for the project;

dollars.cents is the limit of expenditure on this run. If this limit is exceeded, the run will be terminated. The \$ sign and decimal point are optional, and if both are omitted, the cost will be assumed to be in cents. If the \$ sign is present but the decimal point is omitted, the cost will be assumed to be in dollars;

type-of-run is optional, but may be any one of the following:

DEV - development

PROD - production

MAINT - program maintenance

This information will be used for statistical purposes.

The parameters on the .JOB card may be specified using shorthand forms of the identifiers, i.e.

PJ for PROJECT

N " NAME

C " COST

TP " TYPE

Provided either the shorthand or longhand versions of the identifiers is used, the parameters may be keypunched in any order on the .JOB card. For example:

.JOB PROJECT=3176, NAME=JONES, COST=\$3.16

.JOB PJ=4387, C=\$5, N=SMITH, TYPE=DEV

3. The .FORTRAN Card is used for specifying that a FORTRAN IV source language program is to be compiled. It contains the characters ".FORTRAN" in columns 1 through 8. The .FORTRAN card must immediately precede the program to be compiled, and directs the FORTRAN IV compiler to compile the source language statements that follow.

The source program may consist of a main program, and one or more subprograms. The last program or subprogram must be followed by a File Separator Card (see section 5).

The FORTRAN IV compiler translates the source program into relocatable form and produces a source listing on the line printer. The listing will consist of source statements, error comments, and a short program summary.

4. The *.RUN Card* causes all of the previously-compiled relocatable routines to be loaded into memory, library routines to be inserted, and the program executed. The card contains the characters ".RUN" in columns 1 through 4.

The relocatable routines will be destroyed during the loading process, and therefore cannot be subsequently reloaded. If the source program does not contain a main program (i.e. consists only of subprograms), an error message will be printed and the run terminated.

If the program requires input data, then this should immediately follow the .RUN card. Data decks should be terminated by a File Separator Card.

5. The *File Separator Card* is used to signify the end of a program or data file. It contains the file separator character (card code 11-8-9-4) in column 1. The remainder of the card is blank.
6. The *End-of-Job Card* must be physically the last card in a job deck. It consists of the characters .EOJ in columns 1 through 4, and contains no other information.

The .EOJ card signifies the end of a job, and causes the job statistics to be printed. These will contain a summary of processor time used, core memory used, information transfers, etc., and the cost of each of these items. The total cost of the job will be printed, and the balance of the authorized expenditure notified. The summary will be concluded by a message indicating the reason for the termination of the job.

7. Sample Deck Setup

Job Identification Card

.JOB .PROJECT = 4761, NAME = WILCOX, COST = \$1.75, TYPE = DEV

.FORTRAN

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Main Program 1

END

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Subroutine 1

END

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Subroutine 2

END

File Separator Card

.RUN

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Data

File Separator Card

.FORTRAN

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Main Program 2

END

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Subroutine 3

END

File Separator Card

.RUN

⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }
⋮ ⋮ ⋮ }

Data

File Separator Card

.EOJ

